**Balogun Ali**

**17/SCI01/019**

**CSC407**

**Assignment**

**Fallacies of distributed systems**

**1. Network is reliable**

You cannot assume the network is reliable and not worry about network issues. The truth is that networks are more reliable than they used to be. However, they aren’t 100% reliable. When designing and writing your applications, don’t forget to account for network failures.

**2. Latency is zero**

Imagine two applications on the same computer talking to each other. The latency, in this case, will be close to zero, but it won’t be zero. If we introduce a network between the applications, the latency will always be greater than zero.

Latency is an important metric you should be aware of, and monitor for your applications. Latency can have a big impact on user experience and performance.

**3. Infinite bandwidth**

At first, it might seem like there’s plenty of bandwidth. However, when a system has tens or hundreds of services, the amount of communication and data sent back and forth increases significantly. For example, it’s predicted that self-driving cars will produce from 400 GB to 5 TB of data an hour. Design your applications with bandwidth usage in mind.

**4. Network is secure**

This fallacy can be fatal. Security and embracing a defense-in-depth approach must be a priority when designing your applications. It’s not a question if your system will be attacked; it’s a question of when it will be attacked.

**5. Topology doesn’t change**

Indeed, topology doesn’t change when you’re running applications on your computer. But when you deploy the applications to the cloud, the network topology is out of your control. The cloud provider upgrades and changes the network equipment, machines are turned off and new ones are created, and so on. You can’t rely on constant topology in the cloud.

**6. There is one administrator**

In the past, it was common to have a single person responsible for maintaining environments, installing and upgrading applications, and so on. However, that approach has changed with the shift to modern cloud architectures and DevOps practices.

Modern cloud-native applications are composed of many services, working together but developed by different teams. It’s practically impossible for a single person to know and understand the whole application, let alone try to fix all the issues.

Put governance in place that makes it easy to troubleshoot any issues that arise. Concepts such as release management, decoupling, logging, and monitoring apply here.

**7. Transport cost is zero**

You can think about transport costs in two ways:

There’s a cost to network with most cloud providers. Sometimes, the network ingress is free. However, there’s is a cost to the network egress or when you’re moving data out of the cloud providers’ network.

There’s a cost to object serialization and deserialization. Both operations can be expensive in terms of performance.

**8. Network is homogeneous**

Networks are not homogeneous or of the same kind. Instead, networks are heterogeneous. You can’t assume that the network hardware always stays the same. The key point is to focus on standard protocols so that components can communicate, regardless of the hardware.